UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/847,534 | 05/01/2001 | Lev Novik | MS1-694US | 4018 |

22801 7590 06/15/2004

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

| EXAMINER | |
|---|---|
| BULLOCK JR, LEWIS ALEXANDER | |
| **ART UNIT** | **PAPER NUMBER** |
| 2126 | 8 |

DATE MAILED: 06/15/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) |
|---|---|---|
| | 09/847,534 | NOVIK ET AL. |
| | Examiner | Art Unit |
| | Lewis A. Bullock, Jr. | 2126 |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *31 March 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-43* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-43* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *3/4/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

### *Claim Rejections - 35 USC § 102*

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-43 are rejected under 35 U.S.C. 102(e) as being anticipated by WALKER (US 6,138,171).

As to claims 1 and 10, WALKER teaches a computer-implemented method comprising: receiving a plurality of events (message) (col. 9, lines 25-42; col. 10, lines 52-54); applying the plurality of events to a correlation function (FsmInstances), wherein the correlation function is implemented as a state machine and is configured to correlate the plurality of events (via FsmMap or FsmArray) (col. 9, lines 25-57; col. 7, lines 38-41); and generating a specific event (event) if the correlation function is satisfied by the plurality of events (via communication between state machines) (col. 7, lines 47-62).

As to claim 2, WALKER teaches the correlation function is a class object (FsmInstance) (col. 11, line 52 – col. 12, line 13).

As to claim 3, WALKER teaches the steps of: receiving a data element (state identifier / symbolic message identifier); and applying the data element (state identifier / symbolic message identifier) and at least one of the plurality of events (message object) to the correlation function (col. 9, line 47 - col. 10, line 33).

As to claim 4, WALKER teaches the steps of: receiving a plurality of data elements (state identifier / symbolic message identifier) (via a plurality of message objects); and applying the plurality of data elements (state identifiers / symbolic message identifiers in / due to a plurality of message objects) and the plurality of events (message objects) to the correlation function (FsmInstance) (col. 9, line 47 - col. 10, line 33).

As to claim 5, WALKER teaches communicating the specific event (event) to at least one event consumer (state object / another state machine) that subscribed to the specific event (via processEvent member function being called) (col. 11, line 52 - col. 12, line 13; col. 7, lines 42-62).

As to claim 6, WALKER teaches continuing to receive additional events (message objects) and apply the additional events (message objects) to the correlation function (FsmInstance) if the correlation function is not satisfied by the plurality of events (col. 10, lines 52-54; col. 9, lines 25-42).

As to claim 7, WALKER teaches resetting the correlation function (FsmInstance) after generating a specific event (via setStateID member function) (col. 12, lines 4-11).

As to claim 8, WALKER teaches the steps of: creating an instance of a particular state machine (via factory creating FsmInstance); and defining transitions (states) for the particular state machine by subscribing to at least one event (via FsmMap or FsmArray) (col. 11, line 52 – col. 12, line 13; col. 9, lines 25-57).

As to claim 9, WALKER teaches applying an update consumer to the particular state machine to update the state of the particular state machine (via the programmer modifying the initialization files) (col. 6, lines 5-27; col. 7, lines 4-11; col. 8, line 8 – col. 9, line 9).

As to claims 11 and 19, WALKER teaches a computer-implemented method comprising: receiving a plurality of events (messages) (col. 9, lines 25-42; col. 10, lines 52-54); receiving a plurality of data elements (state identifier / symbolic message identifier of messages); identifying a plurality of correlation functions (FsmInstances) configured to correlate the plurality of events and the plurality of data elements; applying the plurality of events (message objects) and the plurality of data elements (state identifier / symbolic message identifier of messages) to the plurality of correlation functions (state identifiers / symbolic message identifiers in / due to a plurality of message objects to the plurality of FsmInstances) (via FsmMap or FsmArray) (col. 9,

lines 25-57; col. 7, lines 38-41); and generating a specific event if at least one of the plurality of correlation functions is satisfied (via communication between state machines or with state machine) (col. 7, lines 47-62; col. 9, lines 25-42).

As to claim 12 and 13, WALKER teaches each of the plurality of correlation functions is implemented as a state machine as an instance of a class object (FsmInstance) (col. 11, line 52 – col. 12, line 13).

As to claim 14, WALKER teaches communicating the specific event (event) to at least one event consumer (state object / another state machine) that subscribed to the specific event (via processEvent member function being called) (col. 11, line 52 - col. 12, line 13; col. 7, lines 42-62).

As to claim 15, WALKER teaches continuing to receive additional events (message objects) and additional data elements (state identifier / symbolic message identifier of messages) and applying the plurality of events with the additional events (message objects), the plurality of data elements with the additional data elements (state identifier / symbolic message identifier of messages) to the correlation functions (FsmInstances) (col. 10, lines 52-54; col. 9, lines 25-42).

As to claim 16, refer to claim 15 for rejection. However, claim 16 further details receiving additional correlation functions and applying the events and data elements to

the additional correlation functions. WALKER teaches creating a plurality of correlation functions (FsmInstances) and applying events (message objects) and data elements (state identifier / symbolic message identifier of messages) to the additional correlation functions (FsmInstances) (col. 11, line 52 – col. 12, line 13).

As to claim 17, WALKER teaches generating the specific event (event) if at least one of the plurality of correlation functions or at least one of the additional correlation functions (FsmInstances) is satisfied (via communication between state machines) (col. 7, lines 47-62).

As to claim 20 and 27, WALKER teaches a computer-implemented method comprising: identifying a schema for creating state machines (FsmInstances) (via the configuration file) (col. 7, line 63 – col. 9, line 3), the state machines (FsmInstances) to correlate at least two events (events the FsmInstances may respond to); creating an instance of a particular state machine (FsmInstances) (via the configuration file) (col. 7, line 63 – col. 9, line 3); defining transitions (state objects) for the particular state machine by subscribing to at least one event (fig. 3); and applying an update consumer to the particular state machine to update the state of the particular state machine (via the programmer modifying the initialization files) (col. 6, lines 5-27; col. 7, lines 4-11; col. 8, line 8 – col. 9, line 9).

As to claims 21 and 22, WALKER teaches the particular state machine includes a timer, and deleting the particular state machine if the timer expires (col. 12, lines 14-36; col. 12, lines 41-65).

As to claims 23 and 24, WALKER teaches the particular state machine (FsmInstance) correlates at least one event (message object) and at least one data element (state object) (col. 9, line 47 - col. 10, line 33).

As to claim 25, WALKER teaches determining a current state of the particular state machine (via current state identifier) (col. 9, lines 38-40).

As to claim 26, WALKER teaches the particular state machine (FsmInstance) is an instance of a class object (col. 11, line 52 – col. 12, line 13).

As to claim 28, WALKER teaches an apparatus comprising: a plurality of event consumers (state objects / another state machine / FsmInstances); and an event correlator (FsmInstance to state objects / FsmMap to FsmInstances / FsmArray to FsmInstances) coupled to the plurality of event consumers, the event correlator to receive events (message objects) from at least one event source and to receive data elements (state identifier / symbolic message identifier of messages) from at least one data source, the event correlator further to receive at least one correlation function (capability of translating message and mapping it the FsmInstance assigned

responsibility for processing the message) configured to correlate events and data elements and to apply the received events (message objects) and the received data elements (state identifier / symbolic message identifier of messages) to the correlation function (FsmInstance) wherein the event correlator generates a specific event (event to be propagated to further dependent state machines) if the received events and the received data satisfy the correlation function (col. 9, lines 25-42; col. 10, lines 52-54; col. 7, lines 38-62; col. 11, line 52 - col. 12, line 13).

As to claim 29, WALKER teaches the event correlator (FsmMap / FsmArray / FsmInstance) communicates the specific event to the plurality of event consumers (delegated FsmInstances / another FsmInstance) (col. 9, lines 25-57; col. 7, lines 47-62).

As to claim 30, WALKER teaches the event correlator (FsmMap / FsmArray / FsmInstance) communicates the specific event (event) to event consumers (delegated FsmInstances) that have requested to receive the specific event (col. 9, lines 25-57; col. 7, lines 47-62).

As to claim 31, WALKER teaches the event correlator (FsmMap / FsmArray) communicates the specific event to a plurality of filters (processEvent function of FsmInstances), wherein each of the plurality of filters is associated with one of the plurality of event consumers (FsmInstances) (col. 12, lines 1-8).

As to claims 32 and 33, WALKER teaches the event correlator (FsmMap / FsmArray / FsmInstance) includes at least one state machine to implement the correlation function (via delegating to a specific FsmInstance to process event) (col. 7, lines 26-41; col. 7, lines 47 – col. 8, line 7; col. 9, lines 25-42).

As to claim 34, WALKER teaches the event correlator (FsmMap / FsmArray / FsmInstance) continues to receive additional events (message objects) and additional data elements (state identifier / symbolic message identifier of messages) and apply the additional events and the additional data elements to the correlation function (FsmInstances) (col. 10, lines 52-54; col. 9, lines 25-42).

As to claims 35-37, reference is made to a computer readable medium that corresponds to the method of claims 11-13 and is therefore met by the rejection of claims 11-13 above.

As to claim 38, WALKER teaches causing the processor to identify a current state of the state machine (via current state identifier) (col. 9, lines 38-40).

As to claim 39, WALKER teaches the steps of: creating a new instance of a state machine to implement a particular correlation function (FsmInstance) (via configuration file); and defining transitions (states) for the new instance of the state

machine by subscribing to at least one event (State defines the behavior of FsmInstance in terms of the events that FsmInstance may respond to while in the current state...)(col. 7, line 63- col. 9, line 3).

As to claims 40-43, refer to claims 28 and 31 for rejection.

***Response to Arguments***

Applicant's arguments filed 3/31/04 have been fully considered but they are not persuasive. Applicant argues that Walker does not teach applying events to a correlation function or generating a specific event as recited in claim 1. The examiner disagrees. Walker teaches the software state machines have the responsibility of processing the received event and determining if a logical state change is required (col. 7, lines 38-41). Walker also teaches that state machines have a functional relationship to one another wherein each state machine is assigned responsibility for a separate set of events and the two state machines interact by sending each other events as required. Communication between the two levels of state machines is via internal events. Therefore, Walker teaches applying events to a correlation function, i.e. a state machine capable of handling the events, and if the correlation function is satisfied, i.e. if a logical state change is required or the cited state machine does not handle the event, generating and sending a specific event, i.e. communicating with another state machine via an internal event. The claims do not set forth what the state machine considers to be satisfied from the received events, therefore any processing using the events can in

essence satisfy the state machine. Therefore, the rejection is maintained as disclosed above.

Applicant argues that Walker's identifiers do not correspond to the data elements in claim 11. The examiner disagrees. The claims do not allude as to what exactly is considered a data element or how it relates to events. In addition, Applicant's response does not allude to what is considered a data element or how Applicant's data element relates to the cited event in the claims. Applicant's response seems to simply argue that the Examiner's version of a data element is not Applicant's which amounts to a general allegation of a patentable limitation. Therefore, the examiner can properly equate the data elements to the identifiers as disclosed in the rejection.

Applicant argues that the Walker does not teach schemas as disclosed in claim 20. The examiner disagrees. Applicant states that the configuration files in Walker are used to create objects, not to create state machines that correlate events. However, Walker teaches that state machines are class instances (col. 7, lines 63 – col. 8, line 6) and all objects of a state machine are created via reference to the configuration file. Therefore, the state machine is created via the configuration file.

Applicant states that the update consumer in claim 20 is a logical component that updates the state of the particular state machine by applying the update consumer to the state machine. The examiner cannot correctly interpret this statement. It seems that Applicant is stating that the limitation of applying an update consumer to the particular state machine to update the state of the particular state machine means the cited update consumer updates the state of the state machine by applying another

update consumer to the state machine. If this interpretation is correct, then in responding the Examiner would like to point out that this limitation is not claimed. However, if the update consumer simply applies itself to the particular state machine, then a user modifying the configuration file which generates the state machine is considered to be an update consumer. There is no language in the claims detailing that the update consumer is a logical component of a computer system. Therefore, such a limitation cannot be considered. Therefore, the rejection is maintained as detailed above.

All of the remaining arguments correspond to the initial argument that Walker does not teach the correlating of events. In Examiner refers to the response provided above and maintains the rejection to the claims as detailed above.

***Conclusion***

3. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

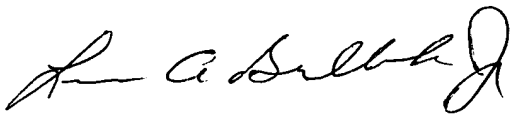
A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (703) 305-0439. The examiner can normally be reached on Monday-Friday, 8:30 am - 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



June 14, 2004